

DICTIONARY OF
COMPUTING

THIRD EDITION

Oxford New York
OXFORD UNIVERSITY PRESS

EXHIBIT A

Oxford University Press, Walton Street, Oxford OX2 6BP

Oxford New York Toronto
Delhi Bombay Calcutta Madras Karachi
Kuala Lumpur Singapore Hong Kong Tokyo
Nairobi Dar es Salaam Cape Town
Melbourne Auckland Madrid
and associated companies in
Berlin Ibadan

Oxford is a trade mark of Oxford University Press

● Market House Books Ltd. 1983, 1986, 1990

First edition 1983

Second edition 1986

Third edition 1990

First issued as an Oxford University Press paperback 1991

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without the prior permission in writing of Oxford University Press. Within the UK, exceptions are allowed in respect of any fair dealing for the purpose of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act, 1988, or in the case of reprographic reproduction in accordance with the terms of the licence issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside these terms and in other countries should be sent to the Rights Department, Oxford University Press, at the address above.

This book is sold subject to the condition that it shall not, by way of trade or otherwise, be lent, re-sold, hired out or otherwise circulated without the publisher's prior consent in any form of binding or cover other than that in which it is published and without a similar condition including this condition being imposed on the subsequent purchaser.

British Library Cataloguing in Publication Data

Data available

Library of Congress Cataloguing in Publication Data

Dictionary of computing.—3rd ed.

p. cm.—(Oxford science publications)

General editor: Valerie Illingworth.

1. Illingworth, Valerie. 2. Computers—Dictionaries.

004.103—dc20 O.A76.15.D526 1990 89-26662

ISBN 0-19-286131-X

5 7 9 10 8 6 4

Printed in Great Britain by
Biddle Ltd
Guildford and King's Lynn

Preface

The great advances in the last few years in all aspects of computing—in theory, technology, and applications—have brought about a tremendous growth in the uses to which computers can be put and in the number of people using them. As the computing scene has expanded so has computing terminology. For this third edition of the *Dictionary of Computing*, over 550 new entries have been added and many of the existing entries have been extensively updated. This reflects recent advances in all aspects of computing, especially new approaches to both programming and computer organization and architecture (and the associated new languages), new developments in the software and hardware associated with microcomputing, networking, and information technology. This edition contains, in single alphabetical listing, nearly 4500 terms used in computing and in the related fields of electronics, mathematics, and logic. The branches of computing covered in this dictionary include:

- algorithms and their properties
- programming languages and concepts
- program development methods
- data structures and file structures
- operating systems and concepts
- computer organization and architecture, past and present
- hardware, including processors, memory devices, and I/O devices
- computer communications
- information technology
- computer applications and techniques
- major computer manufacturers
- legal aspects of computing

The entries in the dictionary have been written by practitioners in these fields of computing and in the associated fields. The terms described range from basic ideas and equipment to advanced concepts of graduate-level computer science; some entries are supplemented by diagrams and notes. The dictionary should be of use to students and teachers of computer science and of all subjects in which computing plays a part. It should also be a valuable reference book to people employed in the various branches of computing as well as to the interested layman with his own micro.

A major undertaking by over fifty people on both sides of the Atlantic, the dictionary has been compiled and prepared for computer typesetting by Oxford House Books Ltd. The editors would like to express their thanks and appreciation to the many contributors for their co-operation, time, and effort.

September 1989

Valerie Illingworth
Edward L. Glaser
I. C. Pyle

*distribution because of the uncertainty in the estimate of the standard deviation (see measures of variation). The probability values depend on an integer f , the number of *degrees of freedom, which is the number associated with the estimate of the standard deviation. Tables of the t distribution are widely available, but algorithms for direct computation are relatively lengthy.

The most common applications are

- (1) testing differences between *means of two samples;
- (2) testing differences from zero of estimated parameters in *regression analysis and *experimental design;
- (3) evaluation of *confidence intervals for means and other estimated quantities.

subgraph A portion of a *graph G obtained by either eliminating edges from G and/or eliminating some vertices and their associated edges. Formally a subgraph of a graph G with vertices V and edges E is a graph G' with vertices V' and edges E' in which V' is a subset of V and E' is a subset of E (edges in E' joining vertices in V').

If V' is a proper *subset of V or E' is a proper subset of E then G' is a *proper subgraph* of G . If all the vertices of G are present in the subgraph G' , then G' is a *spanning subgraph* of G . See also spanning tree.

subgroup A subset T of a *group G on which the dyadic operation \circ is defined; T contains the identity, e , of G , the inverse x^{-1} for any x in T , and the quantity $x \circ y$ for any x and y in T . For any group G the set consisting of e alone is a subgroup; so also is the group G itself. All other subgroups are *proper subgroups* of G .

sublist See list.

submatrix of a given matrix, A . Any matrix derived from A by deleting one

or more of its columns and/or one or more of its rows.

subnet *Short for* communication subnetwork.

sub-Nyquist sampling See Nyquist's criterion.

subprogram Part of a program that may be executed by a *call from elsewhere. The term covers *subroutines, *procedures, and *functions.

subrecursive hierarchy See hierarchy of functions.

subroutine A piece of code that is obeyed "out of line", i.e. control is transferred to the subroutine, and on its completion control reverts to the instruction following the *call. (The instruction code of the CPU usually provides *subroutine jump* and *return* instructions to facilitate this operation.) A subroutine saves space since it occurs only once in the program, though it may be called from many different places in the program. It also facilitates the construction of large programs since subroutines can be formed into libraries for general use. (The same concept appears in high-level languages as the *procedure.)

In the early days of programming, what is now called a subroutine was known as a *closed subroutine*. This was in contrast with the *open subroutine*, which was a piece of code that appeared in several places in a program, and was substituted "in line" by the assembler for each call appearing in the program. The open subroutine was just a convenient shorthand for the programmer; the same facility is now known as a *macro.

subschema See data description language.

subscript A means of referring to particular elements in an ordered collection of elements. For example, if R denotes

such a collection of names then the i th name in the collection may be referenced by R_i (i.e. R subscript i). This printed form is the origin of the term but it is also used when the "subscript" is written on the same line, usually in parentheses or brackets:

$R(i)$ or $R[i]$

See also array.

subsemigroup A *subset T of a *semigroup S , where T is *closed under the dyadic operation \circ defined on S . Let x be an arbitrary element of S . Then the set consisting of

$$x, x \circ x, x \circ x \circ x, \dots$$

i.e. all powers of x , is a subsemigroup of S .

subsequence 1. A *function whose domain is a subset of the positive integers and hence whose image set can be listed:

$$s_1, s_2, \dots, s_m$$

$$\text{where } 1 \leq i_1 < i_2 < \dots < i_m$$

2. The listing of the image set of a subsequence. Hence a subsequence of a string $a_1 a_2 \dots a_n$ is any listing of the form

$$a_{i_1} a_{i_2} \dots a_{i_m}$$

$$\text{where } 1 \leq i_1 < i_2 < \dots < i_m \leq n$$

See also sequence.

subset of a *set S . A set T whose members are all members of S ; this is usually expressed as

$$T \subseteq S$$

A subset T is a *proper subset* of S if there is some element in S that is not in T ; this is expressed as

$$T \subset S$$

substitution A particular kind of mapping on *formal languages. Let Σ_1 and Σ_2 be alphabets. For each symbol, a , in Σ_1 let $s(a)$ be a Σ_2 -language. The function s is a substitution. A *homomorphism occurs where each $s(a)$ is a single word, s is Λ -free if no $s(a)$ contains the empty word.

The function s can be extended to map Σ_1 -words to Σ_2 -languages:

$$s(a_1 \dots a_n) = s(a_1) \dots s(a_n)$$

i.e. the *concatenation of the languages $s(a_1), \dots, s(a_n)$. s can then be further extended to map Σ_1 -languages to Σ_2 -languages:

$$s(L) = \{s(w) \mid w \in L\}$$

$s(L)$ is called the *substitution image* of L under s .

substring of a string of symbols, $a_1 a_2 \dots a_n$. Any string of symbols of the form

$$a_{i_1} a_{i_2} \dots a_{i_j}$$

$$\text{where } 1 \leq i_1 \leq i_2 \leq \dots \leq i_j \leq n$$

The *empty string is regarded as a substring of any string.

substring identifier Let $\alpha = a_1 a_2 \dots a_n$ denote a string in Σ^* and let $\# \in \Sigma$. The substring identifier for position i in α is the shortest substring in α starting at position i that identifies position i uniquely. The existence of such a substring is guaranteed since

$$a_i a_{i+1} \dots a_n \#$$

will always identify position i uniquely. See also position tree.

minuend	0	0	1	1
subtrahend	0	1	0	1
difference	0	1	1	0
borrow	0	1	0	0

Modulo-two subtraction

subtractor An electronic *logic circuit for calculating the difference between two binary numbers, the minuend and the number to be subtracted, the subtrahend (see table). A *full subtractor* performs this calculation with three inputs: minuend bit, subtrahend bit, and borrow bit. It produces two outputs: the difference and the borrow. Full subtractors thus allow for the inclusion of borrows generated by previous stages of subtraction when forming their output